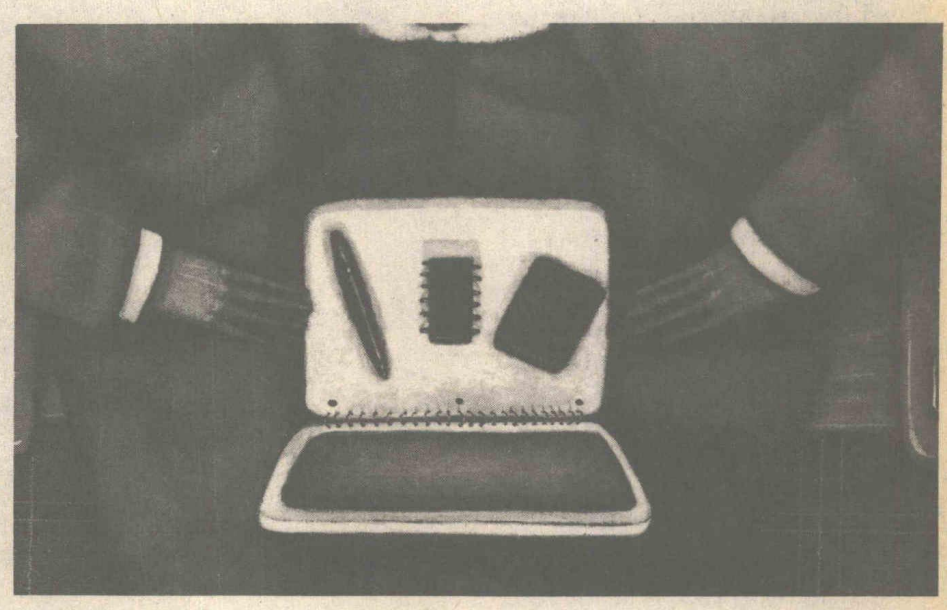
SOFT MAIL

การแปลงภาษาคอมพิวเตอร์

จากปาสคาลเป็นซี

คุณเคยประสบกับปัญหาหรือไม่ ในขณะที่ต้องการ Source โปรแกรมภาษาหนึ่ง แต่คุณมีอีกภาษาหนึ่ง คุณจะทำอย่างไร? ลองศึกษาเทคนิคง่าย ๆ จากเรา อาจจะช่วยแก้ปัญหาของคุณได้



การพัฒนาโปรแกรมคอมพิวเตอร์ในปัจจุบันนี้ ถ้าเราลองนำไปเปรียบเทียบกับการพัฒนาโปรแกรมเมื่อประมาณ 10 กว่าปีที่ผ่านมาหรือมากกว่านั้น จะเห็นว่ามีความแตกต่างกันอย่างมาก โปรแกรมเมอร์ในปัจจุบันนับว่าโชคดีมากกว่าในยุคก่อน ๆ เพราะสามารถที่จะพัฒนาโปรแกรมขึ้นมาได้ง่ายกว่า ทั้งนี้เพราะในปัจจุบันมีฮาร์ดแวร์ที่มีประสิทธิภาพสูงขึ้นและยังมีเครื่องมือหลายอย่างที่ช่วยในการพัฒนาโปรแกรมให้มีความสะดวกรวดเร็วขึ้น ซึ่งผิดกับนักคอมพิวเตอร์ในยุคต้น ๆ ที่จะต้องมีความรู้ทางด้านฮาร์ดแวร์เป็นอย่างดี เพื่อที่จะสามารถเขียนโปรแกรมให้เครื่องทำงานตามคำสั่งได้ ทำให้การพัฒนาโปรแกรมเป็นไปด้วยความลำบากและล่าช้า ผิดกับในปัจจุบันมาก

ในปัจจุบันมีภาษาที่ใช้ในการพัฒนาโปรแกรมมากมาย ในแต่ละภาษามีข้อดีข้อเสียแตกต่างกันไปขึ้นอยู่กับลักษณะงานที่จะนำไปใช้ ดังนั้นก่อนที่จะพัฒนาโปรแกรมใดก็ตาม เราจะต้องศึกษารูปแบบของงานนั้นให้ดีเสียก่อน เพื่อที่จะหาภาษาในการเขียนโปรแกรมให้เหมาะสม

ในแต่ละภาษาที่ใช้เขียนโปรแกรมบางภาษาอาจจะมีความใกล้เคียงกันในด้านรูปแบบหรืออาจแตกต่างกันโดยสิ้นเชิง ในบทความนี้จะเป็นการนำเอาภาษาที่มีรูปแบบของการเขียนที่คล้ายกันซึ่งอาจจะไม่เหมือนกันเลยทีเดียว นำมาจัดรูปแบบใหม่ให้เป็นรูปแบบของอีกภาษาหนึ่ง ซึ่งในบทความนี้จะกล่าวถึงสาเหตุที่ต้องนำมาแปลรูปแบบ การเลือกภาษาที่จะนำมาแปล หลักในการแปล การเปรียบเทียบของภาษาที่จะนำมาแปล ตลอดจนการเขียนโปรแกรมที่จะมาช่วยในการแปลอย่างคร่าว ๆ เพื่อเป็นแนวทางให้ผู้อ่านได้นำหลักการไปใช้และปรับปรุงตามความเหมาะสมต่อไป

เตรียมตัว

ในความเป็นจริงแล้วการที่เราจะสามารถแปลข้อความจากภาษาหนึ่ง (ภาษาทั่วไป) ไปยังอีกภาษาหนึ่งได้ เราจะต้องมีความรู้เกี่ยวกับหลักของภาษาและความหมายของทั้ง 2 ภาษานั้นพอสมควร ยิ่งถ้าเรามีความรู้และคุ้นเคยกับภาษาทั้ง 2 นั้นมากเท่าไร ก็จะยิ่งเป็นการดี เพราะจะได้ผลที่มีความถูกต้องชัดเจนยิ่งขึ้น

ภาษาทางด้านคอมพิวเตอร์ก็เช่นเดียวกัน ยังต้องอาศัยหลักการเดียวกันกับที่กล่าวมาข้างต้น คือ ผู้แปลต้องมีความรู้และความคุ้นเคยกับภาษาต้นทาง (SOURCE) และภาษาปลายทาง (TARGET) และถ้าต้องการความถูกต้องชัดเจนด้วยแล้ว จึงมีความจำเป็นอย่างยิ่งที่จะต้องมีความรู้เกี่ยวกับภาษาทั้ง 2

มากขึ้น

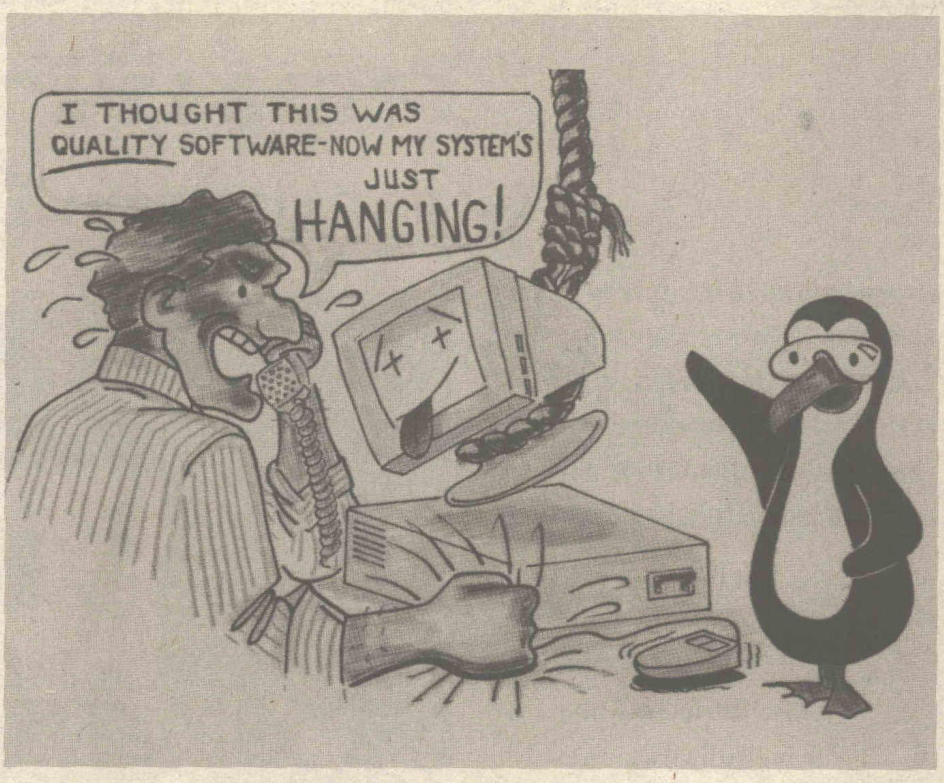
เมื่ออ่านมาถึงตอนนี้ผู้อ่านหลายคนอาจจะเปิดข้ามบทความนี้ไป เพราะไม่คุ้นเคยกับภาษา "การแปลภาษาปาสคาลไปเป็นภาษาซี" ท่านผู้อ่านหลายคนอาจไม่รู้จักรูปแบบของภาษาทั้ง 2 นี้เลย บางคนอาจรู้จักเพียงภาษาใดภาษาหนึ่ง หรือบางคนอาจจะเคยผ่านทั้ง 2 ภาษามาบ้างแล้ว แต่ยังเข้าใจไม่ดีพอ ก็เลยคิดว่าคงอ่านแล้วไม่รู้เรื่อง แต่โดยความจริงแล้วบทความนี้ไม่ได้มุ่งเน้นข้อละเอียดปลีกย่อยของภาษาซีและภาษาปาสคาลลงไปมากมาย เป็นเพียงบอกโครงสร้างหลัก ๆ ที่คล้ายคลึงและแตกต่างของทั้ง 2 ภาษา ดังนั้นผู้ที่ไม่เคยใช้ภาษาทั้ง 2 นี้ ในการโปรแกรมก็ยังสามารถอ่านได้หรือทำความเข้าใจเกี่ยวกับหลักการแปลแล้วนำไปประยุกต์ใช้กับภาษาที่ตนคุ้นเคยได้

แต่ถ้าผู้อ่านท่านใดมีความรู้เกี่ยวกับทั้ง 2 ภาษานี้มาบ้างแล้ว ก็จะทำความเข้าใจและนำไปประยุกต์ใช้ได้ดียิ่งขึ้น

ทำไมต้องแปล

ภาษาพูดและภาษาเขียนที่เราใช้สื่อความหมายโดยทั่วไปนั้น มีอยู่ด้วยกันมากมายหลายภาษา ถ้ามีการใช้ภาษาต่างกันเช่นนี้ การที่จะสื่อความหมายระหว่างคนที่ใช้ภาษาต่างกัน ย่อมต้องมีขั้นตอนที่เป็นตัวกลางในการเชื่อมความหมาย ซึ่งเราเรียกกันว่า "การแปล" (translation)

ในภาษาคอมพิวเตอร์ก็เช่นเดียวกัน ได้มีผู้กำหนดรูปแบบของภาษาไว้มากมาย ภาษาแต่ละภาษามีรูปแบบและประสิทธิภาพต่างกัน เพราะฉะนั้นการแปลจึงเข้ามามีบทบาทในส่วนนี้ ประโยชน์ที่ได้รับจากการแปลภาษา ซึ่งพอจะกล่าวได้ดังนี้



1. เพื่อประสิทธิภาพการใช้งานที่ดีกว่า ภาษาของคอมพิวเตอร์นั้นแต่ละภาษามีข้อกำหนดและประสิทธิภาพต่างกันในด้านการใช้งาน บางภาษานั้นจะมีความง่ายในการอ่านหรือเขียน แต่มีข้อเสียตรงที่ไม่สามารถทำให้เครื่องทำงานได้รวดเร็ว บางภาษาอาจจะเหมาะกับงานทางด้านการโปรแกรมเกี่ยวกับธุรกิจ แต่เมื่อนำมาเขียนเพื่อให้ทำงานทางด้านกราฟิก อาจทำได้ไม่ดีเท่าอีกภาษาหนึ่ง ดังนั้นการที่เราจะเขียนโปรแกรมงานใดงานหนึ่งขึ้นมา เราจะต้องดูลักษณะของงานนั้น ๆ แล้วนำมาพิจารณาว่าเราจะใช้ภาษาใดในการเขียน เพื่อให้ได้งานที่ตรงตามเป้าหมายมากที่สุด ถ้าเรามีตัวโปรแกรมของอีกภาษาหนึ่งอยู่แล้ว แต่เราเห็นว่าภาษาที่ใช้อยู่ไม่มีประสิทธิภาพพอในการทำงานให้ได้ตามเป้าหมาย เราก็สามารถที่จะใช้การแปลเข้ามาช่วยให้เป็นอีกภาษาหนึ่ง ซึ่งเป็นภาษาที่คิดว่าจะทำให้งานของเรามีประสิทธิภาพสูงขึ้น

2. เพื่อการแก้ไขเพิ่มเติมเป็นไปได้โดยสะดวกและรวดเร็ว ภาษาคอมพิวเตอร์นั้นบางภาษาก็มีผู้นิยมใช้กันมาก บางภาษาก็แทบจะไม่มีผู้นิยมใช้เลย ถ้ามีโปรแกรมของงาน ๆ หนึ่งเป็นภาษาที่มีผู้นิยมใช้กันน้อย ผู้ที่จะมาแก้ไขเพิ่มเติมตัวโปรแกรมได้มีอยู่น้อย การพัฒนาแก้ไขโปรแกรมจึงเป็นไปด้วยความยากลำบาก เพื่อความสะดวกในส่วนนี้ เราจึงควรทำการแปลภาษาที่ใช้โปรแกรมนั้นให้เป็นอีกภาษาหนึ่ง ซึ่งมีบุคลากรที่สามารถจะพัฒนาแก้ไขได้ จึงจะง่ายกว่าการที่จะต้องมานั่งเขียนโปรแกรมกันใหม่ ซึ่งใช้เวลามากกว่าหลายเท่าตัว

3. อุปกรณ์การใช้งานที่มีอยู่ไม่สามารถซัพพอร์ตกับภาษาที่นำมาเขียนโปรแกรมได้ ในข้อนี้อาจเกิดขึ้นได้

ต้นฉบับไม่มีหน้า 281-284

No This Page 281-284 In Original

รูปที่ 4 (ต่อ)

```
void checkpart();
void translate(char *s);

/*=============================================================*/

main(int argc,char *argv[])
{
     FILE  *input,*output;
     char   ch[10000];          /* BUFFER ของตัวอักษรใน TEXT.FILE ที่ */
                                /* อ่านเข้ามา                        */

     p=ch;
     if(argc!=3){
     printf("Enter : Convert inputfile outputfile\n");
     exit(1);
     };

     if((input=fopen(argv[1],"r"))==NULL){
     printf("Cannot open inputfile\n");
     exit(1);

     };

     if((output=fopen(argv[2],"w"))==NULL){
     printf("Cannot open outputfile\n");
     exit(1);
     };

     while(!feof(input)){          /* อ่านค่าจาก input file เข้ามาทีละตัว */
             *p++=fgetc(input);
                        };
     p=ch;
     while(*p!=EOF){
     checkpart(p);                 /* ตรวจสอบตัวอักษรจาก input */
    if(tok_type==IDENTIFIER)  translate(token);   /* แปลเมื่อคำที่ตรวจสอบ */
                               /* อยู่ในกลุ่มคำที่กำหนดไว้ใน array */
     fputs(token,output);      /* นำออกทาง output file */
     };

     fclose(input);
     fclose(output);
}
```

รูปที่ 4 (ต่อ)

```
/*=================================================================*/

void checkpart()          /* function ตรวจสอบตัวอักษรจาก input */
{
     char *part;
     part = token;

   if(*p=='\n'){
                    *part++='\r';
                    *part++='\n';
                    *part  ='\0';
                    p++;
                    tok_type = DELIM;
                    return;
                };
    if(*p=='\0'){
                    *part  ='\0';
                    tok_type = DELIM;
                    return;
                };
    while(isspace(*p)) ++p;    /* skip over white space */

    if(*p=='='){               /* เครื่องหมายเปรียบเทียบ */
                    p++;
                    strcpy(token,"==");
                    tok_type = OP;
                    return;
                };

    if(*p==':'){               /* เครื่องหมายให้ค่า */
                    p++;
                    if(*p=='='){
                                    strcpy(token,"=");
                                    p++;
                                }
     else *part++=' ';
                    tok_type = OP;
                    return;
                };
```

รูปที่ 4 (ต่อ)

```
if(strchr("'",*p)){          /* input เป็น string */
                    *part++ = '"';
                    p++;
                    while(!strchr("'",*p)) *part++ = *p++;
                    *part = '"';
                    part++;
                    *part ='\0';
                    p++;
                    tok_type = STRING;
                    return;
                    };


if(strchr("+-*:.,/^%()",*p)){        /* input เป็นเครื่องหมาย */
                                     /* ทางคณิตศาสตร์ */
                    *part = *p;
                    p++;
                    if(*part=='.') *part = ' ';
                    part++;
                    *part = '\0';
                    tok_type = OP;
                    return;
                              };

if(isalpha(*p)){                     /* input เป็นตัวอักษร */
                    while(isalpha(*p)) *part++ = *p++;
                    *part = '\0';
                    tok_type = IDENTIFIER;
                    return;
                 };
if(isdigit(*p)){                     /* input เป็นตัวเลข */
                    while(!isdelim(*p)) *part++ = *p++;
                    tok_type = NUMBER;
                    *part = '\0';
                    return;
                 };
p++;
}
```

รูปที่ 4 (ต่อ)

```
/*=================================================================*/


isdelim(char c)
{
     if(strchr(":;,+-/%()",c) || c=='\r' )
       return 1;
     return 0;
}
/*=================================================================*/
void translate(char *s)

{
    register int i;

      for (i=0;*trans[i][0];i++){
            if(!strcmp(s,trans[i][0])){
                                  strcpy(s,trans[i][1]);
                                  return;
                                      };
                                };
/*=================================================================*/
}
```

อยู่ในรูปแบบของเท็กซ์ไฟล์ของภาษาซีโดยสมบูรณ์ก่อน

แต่จากตัวอย่างในรูปที่ 4 ที่แสดงจะเห็นได้ว่า ตัว output ที่ได้จะช่วยทุ่นแรงและเวลาในการแปลงได้มากทีเดียว

ยังมีจุดหนึ่งที่ผมได้กล่าวไว้ในตอนต้น แต่ยังมิได้บอกแนวทางในการแก้ไขไว้ในตอนนี้จึงถือโอกาสแนะนำแนวทางเล็ก ๆ น้อย ๆ ซึ่งผู้อ่านอาจใช้วิธีอื่นแก้ไขก็ได้ ถ้าเห็นว่าเหมาะสม

ในจุดที่ว่านั้นก็คือ เรื่องการประกาศตัวแปร จากตัวอย่าง output ได้เราจะเห็นว่าส่วนที่เป็นเรื่องของการประกาศตัวแปร ไม่ว่าจะเป็นแบบปกติหรือเป็นแบบการประกาศรับค่าของฟังก์ชันรูปแบบที่ได้ยังมีการสลับกันอยู่เมื่อเทียบกับรูปแบบในภาษาซี เช่น ถ้าในเท็กซ์ไฟล์ของภาษาปาสคาลเป็น

```
X : integer ;
```

เมื่อถูกเอ็กซีคิวต์ โดยโปรแกรมที่เขียนแล้วจะได้ output เป็น

```
X int ;
```

ซึ่งยังไม่ใช้รูปแบบที่ถูกต้องในภาษาซี เพราะประกาศชนิดและชื่อสลับกัน

เมื่อเป็นเช่นนี้เราจึงต้องนำเทคนิคบางอย่างเข้ามาช่วย ซึ่งผู้เขียนจะขอแนะนำการใช้การเก็บค่าเอาไว้ในตัวแปรที่เราสร้างขึ้นเพื่อมารองรับค่านั้นเข้ามาช่วย เมื่อเราอ่านข้อความที่เข้ามาจาก input file แล้ว ตรวจสอบได้ว่าเป็นการประกาศตัวแปรจะนำชื่อของตัวแปร (ซึ่งถูกแสดงไว้ก่อนในภาษาปาสคาล) ใส่ลงในตัวแปรที่รองรับค่านั้น จากนั้นก็อ่านค่าต่อไปเข้ามา ซึ่งค่านี้จะเป็นชนิดของตัวแปรที่กำหนดในภาษาปาสคาล

ต้นฉบับไม่มีหน้า 289

No This Page 289 In Original